
1 Schleifen

Implementieren Sie zur Ausgabe der Fibonacci-Folge $F_{fib} = \{0, 1, 1, 2, 3, 5, \dots\}$ von 0 bis einschließlich 21 drei Varianten. Benutzen Sie dafür jeweils eine `while`-, `do while`- und `for`-Schleife.

2 **If vs. Switch**

Implementieren Sie zwei Programme zum Übersetzen der Zahlen 0–9 einmal mittels `if-else` und einmal mittels `switch`.

```
$ ./program.x 1  
eins  
$ ./program.x 2  
zwei  
...
```

Bonus: Beschreiben Sie den Aufbau der Programme in Assembler-Code und vergleichen Sie den Assembler-Code, den der Compiler für diese generiert. Der Compiler bietet dazu spezielle Compile-Flags wie z. B. `-Os -S` an. Die Manpage von GCC (`man gcc`) bietet einen Überblick über alle Optionen. Alternativ können sie hierfür auch <https://godbolt.org/> verwenden.

3 **Formatierte Ausgabe**

Schreiben Sie ein C-Programm `temperatur.c`, das eine Temperatur in Grad Celsius verarbeitet und verschiedene Informationen dazu ausgibt. Implementieren Sie hierfür die folgenden zwei Funktionen (verwenden Sie keine existierende Konvertierungsfunktion aus einer Mathe-Bibliothek):

```
1 double celsius_to_fahrenheit(double c);  
2 double celsius_to_kelvin(double c);
```

Die Temperatur soll beim Programmaufruf übergeben werden und die verschiedenen Werte dann wohl-formatiert auf der Kommandozeile ausgegeben werden.

```
$ ./temperatur.x 25.3  
Celsius:    25.30 °C  
Fahrenheit: 77.54 °F  
Kelvin:     298.45 K
```

Hinweis: Für die Konvertierung der Text-Eingabe in eine Gleitkommazahl können Sie z.B. `double atof(const char* str);` benutzen.

Material

Makefile und Code-Vorlagen liegen dem beiliegenden Übungsmaterial bei. Sie können alle Binaries automatisch mittels `make` oder gezielt mit `make BINARYNAME` bauen lassen.